Amendments to the Specification

Please amend the specification as follows.

Please amend the title as follows:

~~SEMICONDUCTOR MEMORY CARD, PLAYBACK APPARATUS, RECORDING APPARATUS, PLAYBACK METHOD, RECORDING METHOD, AND A COMPUTER-READABLE STORAGE MEDIUM~~ AUDIO PLAYBACK APPARATUS AND METHOD FOR RESUMING INTERRUPTED PLAYBACK RECORDING

Please amend the paragraph beginning at page 1, line 6, as follows:

This application is based on application Nos. H11-149893, H11-236724, and H11-372605 filed in Japan, the contents of which ~~is~~ are hereby incorporated by reference.

Please amend the paragraph beginning at page 3, line 22, as follows:

The following two methods are also sometimes used to indicate a position at which playback should commence. A third method has the user indicate a move of the playback position to a desired start time within a desired track using a forward or backward search function provided by a playback apparatus. A fourth method has the user indicate a desired track and a desired position within this track using a jog dial (or the like) and then commences reproduction from this position. Since both methods have the user indicate how far the playback previously progressed, they have the same drawback as the second method described above.

Please amend the paragraph beginning at page 12, line 16, as follows:

FIGs. 26A and 26B ~~shows~~ show how an AOB, an AOB_ELEMENT, and an AOB_FRAME that correspond to an arbitrary playback time code are specified;

Please amend the paragraph beginning at page 12, line 24, as follows:

FIGS. 29A and 29B shows how the TKIs are set when two tracks are combined to produce a new track;

Please amend the paragraph beginning at page 13, line 15, as follows:

FIG. 32C shows a pattern where ~~a~~ Type1 and Type2 ~~AOB~~ AOBs are present at the end of a first track and a Type1 AOB is present at the start of a next track;

Please amend the paragraph beginning at page 13, line 18, as follows:

FIG. 32D shows a pattern where ~~a~~ Type1 and Type2 ~~AOB~~ AOBs are present at the end of a first track and ~~a~~ Type2 and ~~a~~ Type1 ~~AOB~~ AOBs ~~is~~ are present at the start of a next track;

Please amend the paragraph beginning at page 15, line 7, as follows:

FIGS. 47A and 47B ~~shows~~ show how a track is divided;

Please amend the paragraph beginning at page 15, line 21, as follows:

~~FIG.~~ FIGS. 54A and 54B ~~shows~~ show how areas in the double buffer 15 are cyclically allocated using ring pointers;

Please amend the paragraph beginning at page 18, line 14, as follows:

The length of a reference number shows the level of the topic in the hierarchy. As a specific example, the number x1 is the number of a drawing that is being referred to in the explanation. The drawings attached to this specification have been numbered in the order in which they are referred to in the specification, so that the order of the drawings roughly matches the order of the explanation. The explanation of certain drawings has been divided into sections, with the reference number x2 giving the section number of a section in the explanation of a drawing indicated by the reference number x1. The reference number x3 shows the number of an additional drawing that is provided to show the details of the section indicated by the section number x2. Finally, the reference number x4 shows the number of a section in the explanation of this additional drawing.

Please amend the paragraph beginning at page 20, line 8, as follows:

~~{3-2} Physical Layer of the Flash Memory Card 31~~

~~The following describes the physical layer of the flash memory card 31. The flash memory is composed of a plurality of sectors, each of which stores 512 bytes of digital data. As one example, a 64MB flash memory card 31 will have a storage capacity of 67,108,864 (=64*1,024*1,024) bytes, so that this card will include 131,072 (=67108864/512) valid sectors. Once the number of replacement sectors, which are provided for use in case of errors, is subtracted, the remaining number of valid sectors into which various kinds of data can be written is around 128,000.~~

{3-2} Physical Layer of the Flash Memory Card 31

The following describes the physical layer of the flash memory card 31. The flash memory is composed of a plurality of sectors, each of which stores 512 bytes of digital data. As one example, a 64MB flash memory card 31 will have a storage capacity of 67,108,864 (=64*1,024*1,024) bytes, so that this card will include 131,072 (=67108864/512) valid sectors. Once the number of replacement sectors, which are provided for use in case of errors, is subtracted, the remaining number of valid sectors into which various kinds of data can be written is around 128,000.

Please amend the paragraph beginning at page 23, line 5, as follows:

~~FIG. 4B shows the construction of the authentication region and the user region in the file system layer. As shown in FIG. 4B, the authentication region and the user region in the file system each include "partition boot sectors", a "file allocation table (FAT)", a "root directory", and a "data region", meaning that the authentication region and the user region have the same construction. FIG. 5 shows the various parts of these file systems in more detail. The following describes the construction of the user region with reference to FIGS. 4A, 4B and 5.~~

FIG. 4B shows the construction of the authentication region and the user region in the file system layer. As shown in FIG. 4B, the authentication region and the user region in the file system each include "partition boot sectors", a "file allocation table (FAT)", a "root directory", and a "data region", meaning that the authentication region and the user

region have the same construction. FIG. 5 shows the various parts of these file systems in more detail. The following describes the construction of the user region with reference to FIGS. 4A, 4B and 5.

Please amend the paragraph beginning at page 28, line 9, as follows:

{3-3} Overview of the Application Layer

in the Flash Memory Card 31

An overview of the application layer in the flash memory card 31 is shown in FIG. 3. As shown by the arrow ~~PN2~~ PN1 drawn with a broken line in FIG. 3, the application layer in the flash memory card 31 is composed of presentation data and navigation data that is used to control the playback of the presentation data. As shown by the arrow PN2, the presentation data includes sets of audio objects (AOB sets) that are produced by encoding audio data that represents music, for example. As shown by the arrow PN3, the ~~The~~ navigation data includes a "PlaylistManager" (PLMG) and a "TrackManager" (TKMG).

Please amend the paragraph beginning at page 31, line 23, as follows:

The "File Key Entries #1, #2, #3, . . . #8" show the first positions of the regions in which the respective FileKeys in the encryption key storing file are stored. Meanwhile, the filenames of AOB files are assigned the serial numbers "001", "002", "003", "004" . . . . These serial numbers show the positions of the corresponding FileKeys in the encryption key sequence, so that the FileKey that was used to encrypt each AOB file will be present in the "FileKey Entry" with the same serial number. In FIG. 9, the arrows Ak1, Ak2, Ak3, . . . show the correspondence between AOB files and FileKeys. In other words, the file "AOB001.SA1" corresponds to the FileKey whose storage position is indicated by the "FileKey Entry#2", and the file "AOB003.SA1" corresponds to the FileKey whose storage position is indicated by the "FileKey Entry#3". As can be understood from rule (3), different FileKeys are used to encrypt different AOB files, with these FileKeys being stored in "FileKey Entries" with the serial numbers "001", "002", "003", "004", etc., given in the filenames of the corresponding AOB files.

Please amend the paragraph beginning at page 34, line 9, as follows:

It should be noted that audio objects can only be compressed according to MPEG2-AAC using the parameters in the parameter table shown in FIG. 11A that is defined in ISO/IEC13818-7. This parameter table is composed of a "Parameter" column, a "Value" column, and a "Comment" column.

Please amend the paragraph beginning at page 68, line 4, as follows:

{17-5_22-15_26A} Use of the Number of Frames given in each AOB_ELEMENT (part 2)

The number of frames written in the BIT is also used when the playback apparatus performs a time search function where playback starts at a point indicated using a time code. ~~In~~ FIG. 26A~~,~~ shows how a playback apparatus can specify the AOB_ELEMENT and AOB_FRAME corresponding to the playback start time indicated by the user. When playback is to commence from a time indicated by the user, the indicate time (in seconds) is set in the Jmp_Entry field, the playback should begin from an AOB_ELEMENT#Y and an AOB_FRAME position x that satisfy Equation 2 given below.

Please amend the paragraph beginning at page 82, line 17, as follows:

FIG. 36 shows how the BIT is set when an AOB is divided as shown in ~~FIG. 35~~ FIGS. 35A and 35B. The AOB shown in ~~FIG. 35~~ FIGS. 35A and 35B is divided at the boundary bd1. The AOB#1 produced by this division includes the two AOB_ELEMENTs AOB_ELEMENT#1 and AOB_ELEMENT#2, while the other AOB#2 produced by this division includes the three AOB_ELEMENTs, AOB_ELEMENT#1, AOB_ELEMENT#2, and AOB_ELEMENT#3.

Please amend the paragraph beginning at page 88, line 9, as follows:

The DPLI referred to here differs from each PLI in the following way. While the DPLI has to indicate all of the tracks stored in the flash memory card ~~52~~ 31, a PLI does not have this restriction and can indicate any number of the tracks. This opens up various possibilities for the user. As representative examples, the user can generate

Playlist_Information indicating only his (her) favorite tracks and store this Playlist_Information in the flash memory card 31, or can have a playback apparatus automatically generate Playlist_Information that only indicates tracks of a certain genre, out of a plurality of tracks stored in the flash memory card 31, and store the resulting Playlist Information in the flash memory card 31.

Please amend the paragraph beginning at page 88, line 24, as follows:

{17-7_18} Number of Playlists and Their Data Sizes

As shown in FIG. 18, a maximum of 99 Playlists can be stored on one flash memory card 31. The combined data size of the PlaylistManager_Information (PLMGI) and the Default Playlist Information (DPLI) is also fixed at 2,560 bytes. Each PLI has a fixed length of 512 bytes. The "DPL_TK_SRP" included in the Default Playlist Information includes a "DPL_TK_ATR" and a "DPL_TKIN". On the other hand, the "PL_TK_SRP". The format of the DPL)TK_ATR, DPL_TKIN, and PL_TKIN fields is shown in ~~FIG. 39~~ FIGS. 39A and 39B.

Please amend the paragraph beginning at page 105, line 15, as follows:

{48-1} External Appearance of the Playback Apparatus

FIG. 48 shows a portable playback apparatus for the flash memory card 31 of the present invention. The playback apparatus shown in FIG. 48 has an insertion slot for inserting the flash memory card 31, a key panel for receiving user indications for operations such as playback, forward search, backward search, fast forward, rewind, stop, etc., and an LCD (liquid crystal display) panel. In terms of appearance, this playback apparatus resembles other kinds of portable music players.

Please amend the paragraph beginning at page 106, line 15, as follows:

an "Audio" key for receiving user selections of the sampling frequency or of stereo or monaural ~~is~~ to be used;

Please amend the paragraph beginning at page 107, line 4, as follows:

(1) A list of playlist and tracks is shown on the LCD panel to allow the user to indicate the ~~Default_Playlist_Information~~ Default_Playlist_Information, a PLI, or separate tracks.

Please amend the paragraph beginning at page 112, line 23, as follows:

{52-5_53_54A,B} Input and Output by the Double Buffer 15

FIG. 53 shows how input and output are performed for the double buffer 15. FIGS. 54A and 54B show how regions in the double buffer 15 are cyclically secured for storing cluster data using a ring ~~pointers~~ pointer.

Please amend the paragraph beginning at page 113, line 25, as follows:

In the present case, the cluster data 002 and 003 ~~is~~ are stored in the double buffer 15 and the read positions ①②③④ are successively indicated by the read pointer, as shown in FIG. 53. When the read pointer reaches the read position ⑤, all of the AOB_FRAMEs included in cluster 002 will have been read, so that cluster 004 is read and, as shown by the arrow w6 in FIG. 54A, is overwritten into the region that was previously occupied by cluster 002.

Please amend the paragraph beginning at page 169, line 5, as follows:

The DPLI_RSM_PL (PLI_RSM_PL) only ~~include~~ includes a Track_Number and Playback_Time, and so differs from the PLMG_RSM_PL in that a Playlist_Number is unnecessary. As another difference, when all of the tracks in the playback order indicated by the DPLI or a PLI have been completely played back, the value "FF" is set in the Track_Number in the DPLI_RSM_PL (PLI_RSM_PL) to show that the playlist was completely played back.

Please amend the paragraph beginning at page 172, line 15, as follows:

Note that since the resumption of playback for every playlist is performed in this embodiment according to the Track_Number and Playback_Time in the PLI_RSM_PL (DPLI_RSM_PL), it is preferable for the user indication of the playlist to be made via a

menu like that shown in FIG. 77 instead of via the menu of the first embodiment shown in FIG. 49 that merely gives a list of the playlists. FIG. 77 shows an example menu that displays the playlists together with the settings of the PLI_RSM_PL for each playlist for the case where the playback ranges (1) to (3) shown in FIG. 76 have already been played back. PLIs that have not had their track sequences played back in ~~its~~ their entirety are displayed with a track number showing the Track_Number in the PLI_RSM_PL and a playback time based on the value of the Playback_Time in the PLI_RSM_PL. Conversely, PLIs that have had their track sequences played back in their entirety have the value "FF" set in the Track_Number in the PLI_RSM_PL and so are displayed with an indication showing that playback is complete. As a result, this menu tells the user how much of each playlist has been played back, so that the user can know which playlists have been entirely played back and which playlists have only been partially played back.

Please amend the paragraph beginning at page 173, line 15, as follows:

While music applications are stored on the flash memory card 31 in the first to third embodiments, the present embodiment relates to an improvement in the storage of short-lived applications. Here a "short-lived application" refers to any application, such as news, an audio magazine, a recording of a speech, etc., that only needs to be listened to once, and so differs from music applications that are repeatedly listened to. As conventional examples of the short-lived applications, magazines tend to be published weekly or monthly while news tends to be published every day.

Please amend the paragraph beginning at page 176, line 22, as follows:

With this embodiment, short-lived applications, such as news, can be downloaded and stored on a flash memory card 31. Such short-lived applications can be deleted starting with the applications that have been entirely played back, so that even when a short-lived application such as news is produced every day, such short-lived applications can be prevented from taking up all of the storage capacity of the flash memory card 31.